

Introducción a Bases de Datos Avanzadas

Bases de Datos Avanzadas

Universidad del Rosario

Maestría MACC

- 1 Sistemas de Manejos de Bases de Datos (DBSM)
- 2 Modelos de Datos
 - Lenguajes para Manipulación de Datos
 - Manejo de Transacciones
- 3 Base de Datos Relacional (RDB)
 - Claves/*Keys*
 - Claves Foráneas/*Foreign Keys*
 - Esquema de una Base de Datos
- 4 Álgebra Relacional
- 5 Ejercicios

Definición de DBSM



Ejemplos:

- DB de universidades (calificaciones, profesores, alumnos, clases).
- RRSS (conexiones entre usuarios -MG, follow/unfollow-).
- Bancos y Finanzas.



Usos de bases de datos:

- **Procesamiento online de transacciones:** muchos usuarios, pocos datos y updates.
- **Análítica de Datos (Data Analytics):** procesos de decisión en negocios.

Definición de DBSM



Ejemplos:

- DB de universidades (calificaciones, profesores, alumnos, clases).
- RRSS (conexiones entre usuarios -MG, follow/unfollow-).
- Bancos y Finanzas.



Usos de bases de datos:

- **Procesamiento online de transacciones:** muchos usuarios, pocos datos y updates.
- **Analítica de Datos (Data Analytics):** procesos de decisión en negocios.

Definición de DBSM



Ejemplos:

- DB de universidades (calificaciones, profesores, alumnos, clases).
- RRSS (conexiones entre usuarios -MG, follow/unfollow-).
- Bancos y Finanzas.



Usos de bases de datos:

- **Procesamiento online de transacciones:** muchos usuarios, pocos datos y updates.
- **Análítica de Datos (Data Analytics):** procesos de decisión en negocios.

Definición de DBSM



Ejemplos:

- DB de universidades (calificaciones, profesores, alumnos, clases).
- RRSS (conexiones entre usuarios -MG, follow/unfollow-).
- Bancos y Finanzas.



Usos de bases de datos:

- **Procesamiento online de transacciones:** muchos usuarios, pocos datos y updates.
- **Análítica de Datos (Data Analytics):** procesos de decisión en negocios.

Definición de DBSM



Ejemplos:

- DB de universidades (calificaciones, profesores, alumnos, clases).
- RRSS (conexiones entre usuarios -MG, follow/unfollow-).
- Bancos y Finanzas.



Usos de bases de datos:

- **Procesamiento online de transacciones:** muchos usuarios, pocos datos y updates.
- **Análítica de Datos (Data Analytics):** procesos de decisión en negocios.

Definición de DBSM



Ejemplos:

- DB de universidades (calificaciones, profesores, alumnos, clases).
- RRSS (conexiones entre usuarios -MG, follow/unfollow-).
- Bancos y Finanzas.



Usos de bases de datos:

- **Procesamiento online de transacciones:** muchos usuarios, pocos datos y updates.
- **Análítica de Datos (Data Analytics):** procesos de decisión en negocios.

Definición de DBSM



Ejemplos:

- DB de universidades (calificaciones, profesores, alumnos, clases).
- RRSS (conexiones entre usuarios -MG, follow/unfollow-).
- Bancos y Finanzas.



Usos de bases de datos:

- **Procesamiento online de transacciones:** muchos usuarios, pocos datos y updates.
- **Análítica de Datos (Data Analytics):** procesos de decisión en negocios.

Definición de DBSM



Ejemplos:

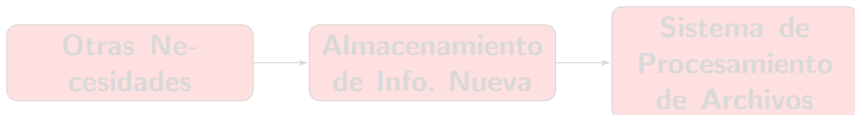
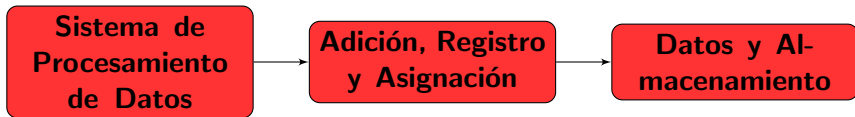
- DB de universidades (calificaciones, profesores, alumnos, clases).
- RRSS (conexiones entre usuarios -MG, follow/unfollow-).
- Bancos y Finanzas.



Usos de bases de datos:

- **Procesamiento online de transacciones:** muchos usuarios, pocos datos y updates.
- **Análítica de Datos (Data Analytics):** procesos de decisión en negocios.

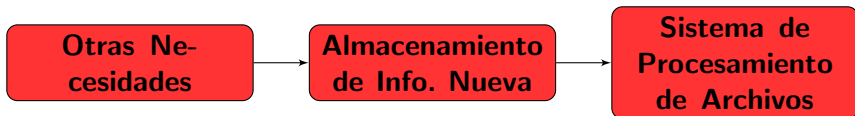
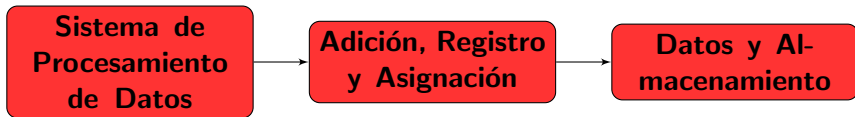
DBs y Sistemas de Procesamiento de Datos



Problemas con el Procesamiento de Archivos.

- Redundancia e inconsistencia en datos.
- Dificultad de acceso.
- Problemas de *Atomicidad*, seguridad, integridad, etc.

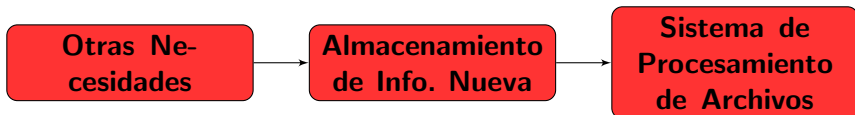
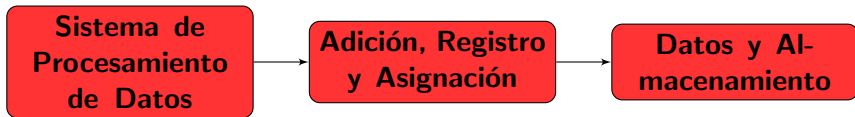
DBs y Sistemas de Procesamiento de Datos



Problemas con el Procesamiento de Archivos.

- Redundancia e inconsistencia en datos.
- Dificultad de acceso.
- Problemas de *Atomicidad*, seguridad, integridad, etc.

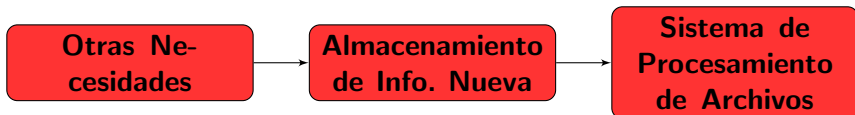
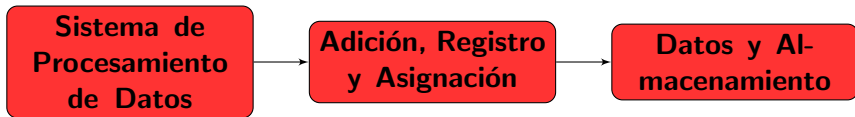
DBs y Sistemas de Procesamiento de Datos



Problemas con el Procesamiento de Archivos.

- Redundancia e inconsistencia en datos.
- Dificultad de acceso.
- Problemas de *Atomicidad*, seguridad, integridad, etc.

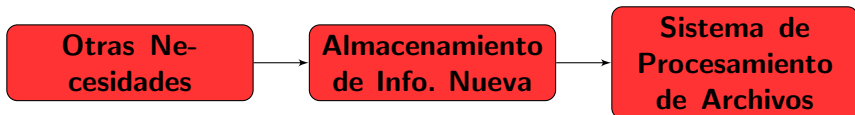
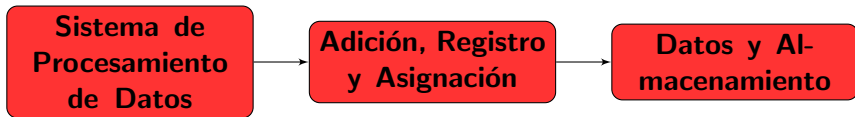
DBs y Sistemas de Procesamiento de Datos



Problemas con el Procesamiento de Archivos.

- Redundancia e inconsistencia en datos.
- Dificultad de acceso.
- Problemas de *Atomicidad*, seguridad, integridad, etc.

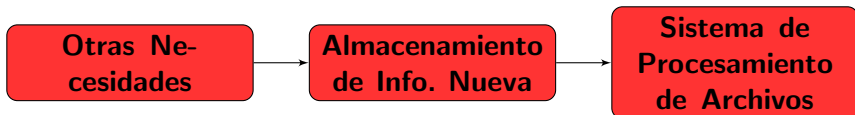
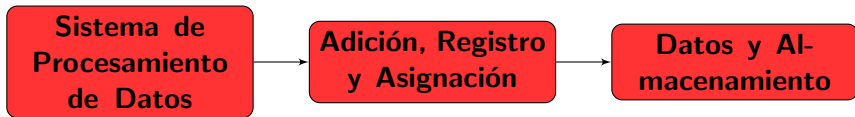
DBs y Sistemas de Procesamiento de Datos



Problemas con el Procesamiento de Archivos.

- Redundancia e inconsistencia en datos.
- Dificultad de acceso.
- Problemas de *Atomicidad*, seguridad, integridad, etc.

DBs y Sistemas de Procesamiento de Datos



Problemas con el Procesamiento de Archivos.

- Redundancia e inconsistencia en datos.
- Dificultad de acceso.
- Problemas de *Atomicidad*, seguridad, integridad, etc.

Tipos de Modelos de Datos

- ❶ **Modelo Relacional:** estructura de tabla.

Identificación	Curso
10001	Programación y Métodos Numéricos
...	...

- ❷ **Modelo Semirrelacional:** tipo *diccionario* (JSON, XML).

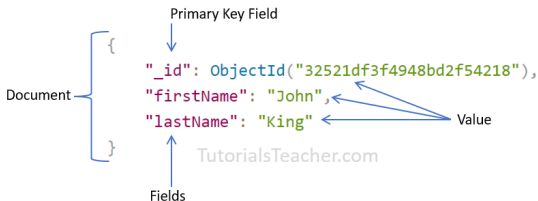
- ❸ **Modelos Basados en Objetos:** programación orientada a objetos (OOP).

Tipos de Modelos de Datos

❶ **Modelo Relacional:** estructura de tabla.

Identificación	Curso
10001	Programación y Métodos Numéricos
...	...

❷ **Modelo Semirrelacional:** tipo *diccionario* (JSON, XML).



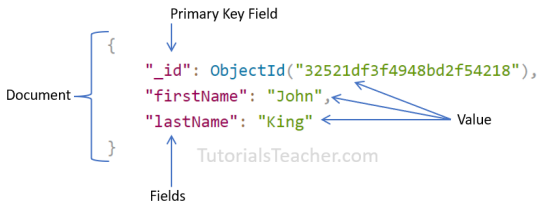
❸ **Modelos Basados en Objetos:** programación orientada a objetos (OOP).

Tipos de Modelos de Datos

- ❶ **Modelo Relacional:** estructura de tabla.

Identificación	Curso
10001	Programación y Métodos Numéricos
...	...

- ❷ **Modelo Semirrelacional:** tipo *diccionario* (JSON, XML).



- ❸ **Modelos Basados en Objetos:** programación orientada a objetos (OOP).

Modelo de Datos Relacional



ID	Apellido	Departamento	Salario (\$ mill)
1001	Díaz	Ciencias de la Computación	10
2001	González	Biología	11
3001	Bonnett	Literatura	12



Departamento	Edificio	Presupuesto (\$ mill)
Ciencias de la Computación	Taylor	500
Biología	M. Cadena	300
Literatura	León de Greiff	200



Niveles de abstracción de datos:

- *Nivel físico:* almacenamiento.
- *Nivel lógico:* datos y sus relaciones.
- *Nivel visual:* simplificación interacción con el sistema.

Modelo de Datos Relacional



ID	Apellido	Departamento	Salario (\$ mill)
1001	Díaz	Ciencias de la Computación	10
2001	González	Biología	11
3001	Bonnett	Literatura	12



Departamento	Edificio	Presupuesto (\$ mill)
Ciencias de la Computación	Taylor	500
Biología	M. Cadena	300
Literatura	León de Greiff	200



Niveles de abstracción de datos:

- *Nivel físico:* almacenamiento.
- *Nivel lógico:* datos y sus relaciones.
- *Nivel visual:* simplificación interacción con el sistema.

Modelo de Datos Relacional



ID	Apellido	Departamento	Salario (\$ mill)
1001	Díaz	Ciencias de la Computación	10
2001	González	Biología	11
3001	Bonnett	Literatura	12



Departamento	Edificio	Presupuesto (\$ mill)
Ciencias de la Computación	Taylor	500
Biología	M. Cadena	300
Literatura	León de Greiff	200



Niveles de abstracción de datos:

- *Nivel físico*: almacenamiento.
- *Nivel lógico*: datos y sus relaciones.
- *Nivel visual*: simplificación interacción con el sistema.

Modelo de Datos Relacional



ID	Apellido	Departamento	Salario (\$ mill)
1001	Díaz	Ciencias de la Computación	10
2001	González	Biología	11
3001	Bonnett	Literatura	12



Departamento	Edificio	Presupuesto (\$ mill)
Ciencias de la Computación	Taylor	500
Biología	M. Cadena	300
Literatura	León de Greiff	200



Niveles de abstracción de datos:

- *Nivel físico*: almacenamiento.
- *Nivel lógico*: datos y sus relaciones.
- *Nivel visual*: simplificación interacción con el sistema.

Modelo de Datos Relacional



ID	Apellido	Departamento	Salario (\$ mill)
1001	Díaz	Ciencias de la Computación	10
2001	González	Biología	11
3001	Bonnett	Literatura	12




Departamento	Edificio	Presupuesto (\$ mill)
Ciencias de la Computación	Taylor	500
Biología	M. Cadena	300
Literatura	León de Greiff	200




Niveles de abstracción de datos:

- *Nivel físico*: almacenamiento.
- *Nivel lógico*: datos y sus relaciones.
- *Nivel visual*: simplificación interacción con el sistema.

Modelo de Datos Relacional



ID	Apellido	Departamento	Salario (\$ mill)
1001	Díaz	Ciencias de la Computación	10
2001	González	Biología	11
3001	Bonnett	Literatura	12



Departamento	Edificio	Presupuesto (\$ mill)
Ciencias de la Computación	Taylor	500
Biología	M. Cadena	300
Literatura	León de Greiff	200

 Niveles de abstracción de datos:

- *Nivel físico*: almacenamiento.
- *Nivel lógico*: datos y sus relaciones.
- *Nivel visual*: simplificación interacción con el sistema.

Propiedades del Modelo de Datos Relacional

I Almacenamiento de datos (atributos):

ID	Apellido	Departamento	Salario (\$ mill)
----	----------	--------------	-------------------

Departamento	Edificio	Presupuesto (\$ mill)
--------------	----------	-----------------------

• *Profesores, Departamentos*

• **Esquema/Schema:** diseño/organización tabla.

• **Instancia/Instance:** información almacenada en cierto momento.

III *ID: char(5)/int; Apellido: char(25); Departamento: char(30); Salario: float(2,3)*

IV *Salario: float(2,3)*

• 2: parte entera. 3: decimales.

Propiedades del Modelo de Datos Relacional

❶ Almacenamiento de datos (atributos):

ID	Apellido	Departamento	Salario (\$ mill)
----	----------	--------------	-------------------

Departamento	Edificio	Presupuesto (\$ mill)
--------------	----------	-----------------------

- *Profesores, Departamentos*
- **Esquema/Schema:** diseño/organización tabla.
- **Instancia/Instance:** información almacenada en cierto momento.

❷ *ID: char(5)/int; Apellido: char(25); Departamento: char(30); Salario: float(2,3)*

❸ *Salario: float(2,3)*

- 2: parte entera. 3: decimales.

Propiedades del Modelo de Datos Relacional

❶ Almacenamiento de datos (atributos):

ID	Apellido	Departamento	Salario (\$ mill)
----	----------	--------------	-------------------

Departamento	Edificio	Presupuesto (\$ mill)
--------------	----------	-----------------------

- *Profesores, Departamentos*
- **Esquema/Schema:** diseño/organización tabla.
- **Instancia/Instance:** información almacenada en cierto momento.

❷ *ID: char(5)/int; Apellido: char(25); Departamento: char(30); Salario: float(2,3)*

❸ *Salario: float(2,3)*

- 2: parte entera. 3: decimales.

Propiedades del Modelo de Datos Relacional

I Almacenamiento de datos (atributos):

ID	Apellido	Departamento	Salario (\$ mill)
----	----------	--------------	-------------------

II

Departamento	Edificio	Presupuesto (\$ mill)
--------------	----------	-----------------------

- *Profesores, Departamentos*
- **Esquema/Schema:** diseño/organización tabla.
- **Instancia/Instance:** información almacenada en cierto momento.

III

ID: char(5)/int; Apellido: char(25); Departamento: char(30); Salario: float(2,3)

IV

Salario: float(2,3)

- 2: parte entera. 3: decimales.

Propiedades del Modelo de Datos Relacional

❶ Almacenamiento de datos (atributos):

ID	Apellido	Departamento	Salario (\$ mill)
----	----------	--------------	-------------------

Departamento	Edificio	Presupuesto (\$ mill)
--------------	----------	-----------------------

- *Profesores, Departamentos*
- **Esquema/Schema:** diseño/organización tabla.
- **Instancia/Instance:** información almacenada en cierto momento.

❷ *ID: char(5)/int; Apellido: char(25); Departamento: char(30); Salario: float(2,3)*

❸ *Salario: float(2,3)*

- 2: parte entera. 3: decimales.

Propiedades del Modelo de Datos Relacional

- ❶ Almacenamiento de datos (atributos):

ID	Apellido	Departamento	Salario (\$ mill)
----	----------	--------------	-------------------

❷

Departamento	Edificio	Presupuesto (\$ mill)
--------------	----------	-----------------------

- *Profesores, Departamentos*
- **Esquema/Schema:** diseño/organización tabla.
- **Instancia/Instance:** información almacenada en cierto momento.

- ❸ *ID: **char**(5)/**int**; Apellido: **char**(25); Departamento: **char**(30); Salario: **float**(2,3)*

- ❹ *Salario: **float**(2,3)*

- 2: parte entera. 3: decimales.

Propiedades del Modelo de Datos Relacional

- ❶ Almacenamiento de datos (atributos):

ID	Apellido	Departamento	Salario (\$ mill)
----	----------	--------------	-------------------

- ❷

Departamento	Edificio	Presupuesto (\$ mill)
--------------	----------	-----------------------

- *Profesores, Departamentos*
- **Esquema/Schema:** diseño/organización tabla.
- **Instancia/Instance:** información almacenada en cierto momento.

- ❸ *ID: **char**(5)/**int**; Apellido: **char**(25); Departamento: **char**(30); Salario: **float**(2,3)*

- ❹ *Salario: **float**(2,3)*

- 2: parte entera. 3: decimales.

Propiedades del Modelo de Datos Relacional

- ❶ Almacenamiento de datos (atributos):

ID	Apellido	Departamento	Salario (\$ mill)
----	----------	--------------	-------------------

- ❷

Departamento	Edificio	Presupuesto (\$ mill)
--------------	----------	-----------------------

- *Profesores, Departamentos*
- **Esquema/Schema:** diseño/organización tabla.
- **Instancia/Instance:** información almacenada en cierto momento.

- ❸ *ID: **char**(5)/**int**; Apellido: **char**(25); Departamento: **char**(30); Salario: **float**(2,3)*

- ❹ *Salario: **float**(2,3)*
 - 2: parte entera. 3: decimales.

Tipos de Lenguajes

- ❶ *Data-Definition Language (DDL).*
- ❷ Definición del *Esquema*. **Ejemplo:** CREATE TABLE en SQL.
- ❸ Ligaduras (*constraints*) en los datos (P. ej., Salario ≥ 0).
 - *Ligadura de Dominio:* tipo de dato.
 - *Integridad Referencial:* entre atributos iguales de DBs.
 - *Autorización:* acceso a diferentes usuarios (lectura, inserción, borrado).
- ❹ *Data-Manipulation Language (DML).*
- ❺ Acceso o manipulación de datos. **Ejemplo:** INSERT y Queries en SQL.
 - Procedimental (especificar la obtención de datos) y Declarativos (sin especificar la obtención de datos).

Tipos de Lenguajes

- ❶ *Data-Definition Language (DDL).*
- ❷ Definición del *Esquema*. **Ejemplo:** CREATE TABLE en SQL.
- ❸ Ligaduras (*constraints*) en los datos (P. ej., Salario ≥ 0).
 - *Ligadura de Dominio:* tipo de dato.
 - *Integridad Referencial:* entre atributos iguales de DBs.
 - *Autorización:* acceso a diferentes usuarios (lectura, inserción, borrado).
- ❹ *Data-Manipulation Language (DML).*
- ❺ Acceso o manipulación de datos. **Ejemplo:** INSERT y Queries en SQL.
 - Procedimental (especificar la obtención de datos) y Declarativos (sin especificar la obtención de datos).

Tipos de Lenguajes

- ❶ *Data-Definition Language (DDL).*
- ❷ Definición del *Esquema*. **Ejemplo:** CREATE TABLE en SQL.
- ❸ Ligaduras (*constraints*) en los datos (P. ej., Salario ≥ 0).
 - *Ligadura de Dominio:* tipo de dato.
 - *Integridad Referencial:* entre atributos iguales de DBs.
 - *Autorización:* acceso a diferentes usuarios (lectura, inserción, borrado).
- ❹ *Data-Manipulation Language (DML).*
- ❺ Acceso o manipulación de datos. **Ejemplo:** INSERT y Queries en SQL.
 - Procedimental (especificar la obtención de datos) y Declarativos (sin especificar la obtención de datos).

Tipos de Lenguajes

- ❶ *Data-Definition Language (DDL).*
- ❷ Definición del *Esquema*. **Ejemplo:** CREATE TABLE en SQL.
- ❸ Ligaduras (*constraints*) en los datos (P. ej., Salario ≥ 0).
 - *Ligadura de Dominio:* tipo de dato.
 - *Integridad Referencial:* entre atributos iguales de DBs.
 - *Autorización:* acceso a diferentes usuarios (lectura, inserción, borrado).
- ❹ *Data-Manipulation Language (DML).*
- ❺ Acceso o manipulación de datos. **Ejemplo:** INSERT y Queries en SQL.
 - Procedimental (especificar la obtención de datos) y Declarativos (sin especificar la obtención de datos).

Tipos de Lenguajes

- ❶ *Data-Definition Language (DDL).*
- ❷ Definición del *Esquema*. **Ejemplo:** CREATE TABLE en SQL.
- ❸ Ligaduras (*constraints*) en los datos (P. ej., Salario ≥ 0).
 - *Ligadura de Dominio:* tipo de dato.
 - *Integridad Referencial:* entre atributos iguales de DBs.
 - *Autorización:* acceso a diferentes usuarios (lectura, inserción, borrado).
- ❹ *Data-Manipulation Language (DML).*
- ❺ Acceso o manipulación de datos. **Ejemplo:** INSERT y Queries en SQL.
 - Procedimental (especificar la obtención de datos) y Declarativos (sin especificar la obtención de datos).

Tipos de Lenguajes

- ❶ *Data-Definition Language (DDL).*
- ❷ Definición del *Esquema*. **Ejemplo:** CREATE TABLE en SQL.
- ❸ Ligaduras (*constraints*) en los datos (P. ej., Salario ≥ 0).
 - *Ligadura de Dominio:* tipo de dato.
 - *Integridad Referencial:* entre atributos iguales de DBs.
 - *Autorización:* acceso a diferentes usuarios (lectura, inserción, borrado).
- ❹ *Data-Manipulation Language (DML).*
- ❺ Acceso o manipulación de datos. **Ejemplo:** INSERT y Queries en SQL.
 - Procedimental (especificar la obtención de datos) y Declarativos (sin especificar la obtención de datos).

Tipos de Lenguajes

- ❶ *Data-Definition Language (DDL).*
- ❷ Definición del *Esquema*. **Ejemplo:** CREATE TABLE en SQL.
- ❸ Ligaduras (*constraints*) en los datos (P. ej., Salario ≥ 0).
 - *Ligadura de Dominio:* tipo de dato.
 - *Integridad Referencial:* entre atributos iguales de DBs.
 - *Autorización:* acceso a diferentes usuarios (lectura, inserción, borrado).
- ❹ *Data-Manipulation Language (DML).*
- ❺ Acceso o manipulación de datos. **Ejemplo:** INSERT y Queries en SQL.
 - Procedimental (especificar la obtención de datos) y Declarativos (sin especificar la obtención de datos).

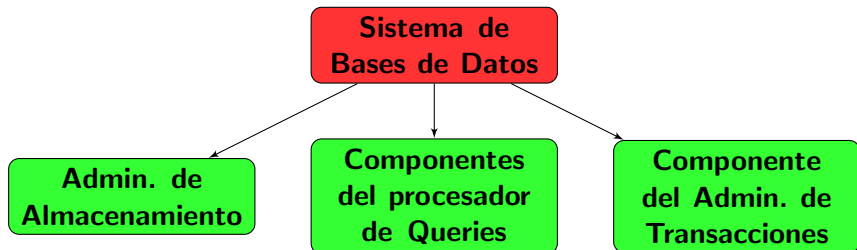
Tipos de Lenguajes

- ❶ *Data-Definition Language (DDL)*.
- ❷ Definición del *Esquema*. **Ejemplo:** CREATE TABLE en SQL.
- ❸ Ligaduras (*constraints*) en los datos (P. ej., Salario ≥ 0).
 - *Ligadura de Dominio*: tipo de dato.
 - *Integridad Referencial*: entre atributos iguales de DBs.
 - *Autorización*: acceso a diferentes usuarios (lectura, inserción, borrado).
- ❹ *Data-Manipulation Language (DML)*.
- ❺ Acceso o manipulación de datos. **Ejemplo:** INSERT y Queries en SQL.
 - Procedimental (especificar la obtención de datos) y Declarativos (sin especificar la obtención de datos).

Tipos de Lenguajes

- ❶ *Data-Definition Language (DDL)*.
 - ❷ Definición del *Esquema*. **Ejemplo:** CREATE TABLE en SQL.
 - ❸ Ligaduras (*constraints*) en los datos (P. ej., Salario ≥ 0).
 - *Ligadura de Dominio*: tipo de dato.
 - *Integridad Referencial*: entre atributos iguales de DBs.
 - *Autorización*: acceso a diferentes usuarios (lectura, inserción, borrado).
- ❹ *Data-Manipulation Language (DML)*.
- ❺ Acceso o manipulación de datos. **Ejemplo:** INSERT y Queries en SQL.
 - Procedimental (especificar la obtención de datos) y Declarativos (sin especificar la obtención de datos).

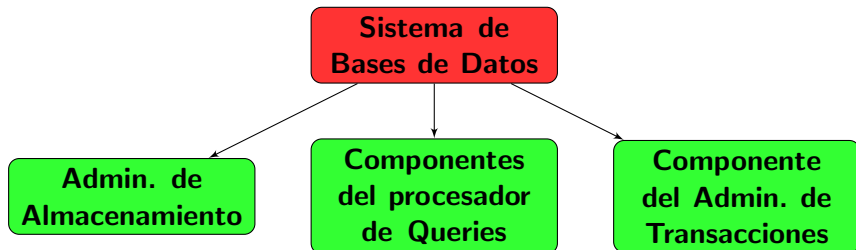
Motor de la Base de Datos



① Procesador de Queries: **acceso a datos**.

② Admin. de Transacciones: **atomicidad**.

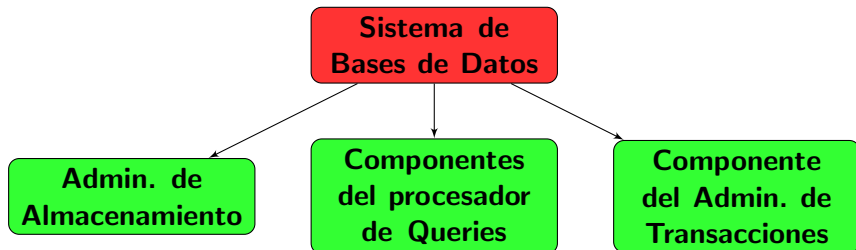
Motor de la Base de Datos



① Procesador de Queries: **acceso a datos**.

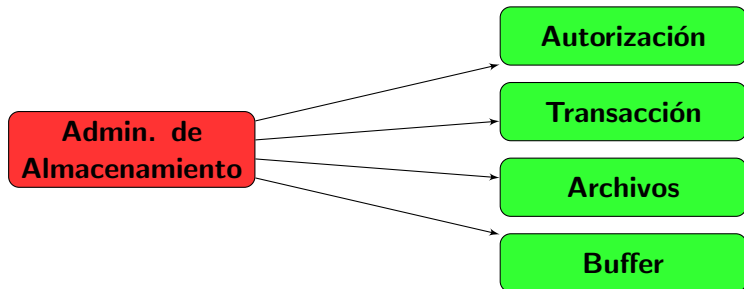
② Admin. de Transacciones: **atomicidad**.

Motor de la Base de Datos



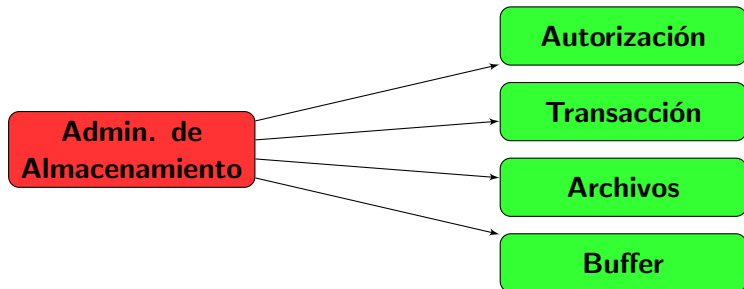
- ❶ Procesador de Queries: **acceso a datos.**
- ❷ Admin. de Transacciones: **atomicidad.**

Admin. de Almacenamiento



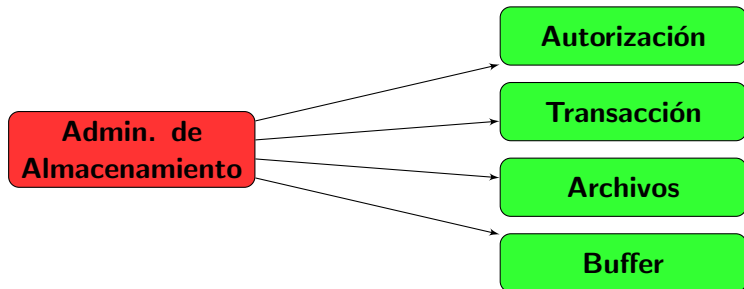
- ❶ Autorización: acceso a datos.
- ❷ Transacción: integridad de la DB.
- ❸ Archivos: localización de espacio de almacenamiento y estructura de datos.
- ❹ Buffer: datos en almacenamiento → datos a memoria principal.

Admin. de Almacenamiento



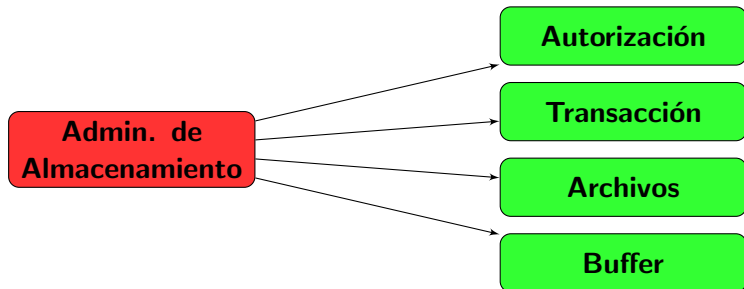
- ❶ Autorización: acceso a datos.
- ❷ Transacción: integridad de la DB.
- ❸ Archivos: localización de espacio de almacenamiento y estructura de datos.
- ❹ Buffer: datos en almacenamiento → datos a memoria principal.

Admin. de Almacenamiento



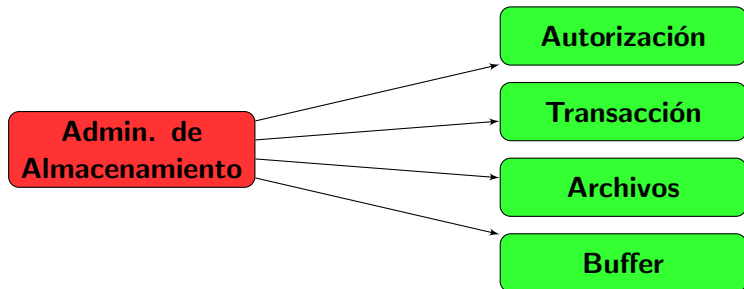
- ❶ Autorización: acceso a datos.
- ❷ Transacción: integridad de la DB.
- ❸ Archivos: localización de espacio de almacenamiento y estructura de datos.
- ❹ Buffer: datos en almacenamiento → datos a memoria principal.

Admin. de Almacenamiento



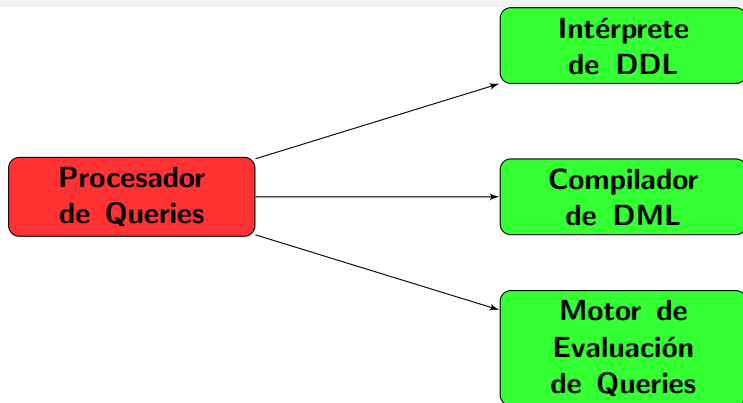
- ❶ Autorización: acceso a datos.
- ❷ Transacción: integridad de la DB.
- ❸ Archivos: localización de espacio de almacenamiento y estructura de datos.
- ❹ Buffer: datos en almacenamiento → datos a memoria principal.

Admin. de Almacenamiento



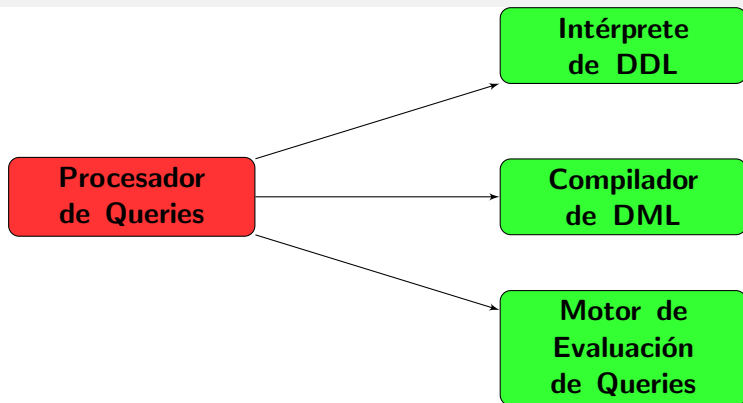
- ❖ Autorización: acceso a datos.
- ❖ Transacción: integridad de la DB.
- ❖ Archivos: localización de espacio de almacenamiento y estructura de datos.
- ❖ Buffer: datos en almacenamiento → datos a memoria principal.

Procesador de Queries



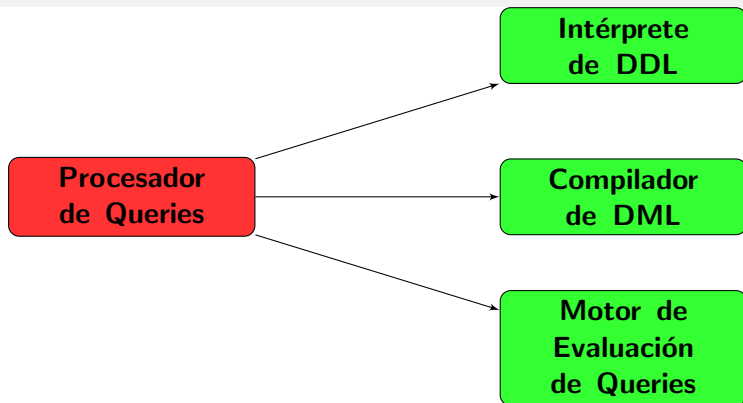
- ❶ Intérprete de DDL: diccionario de datos.
- ❷ Compilador de DML: plan de evaluación.
- ❸ Motor de evaluación: Instrucciones de bajo nivel del compilador de DML.

Procesador de Queries



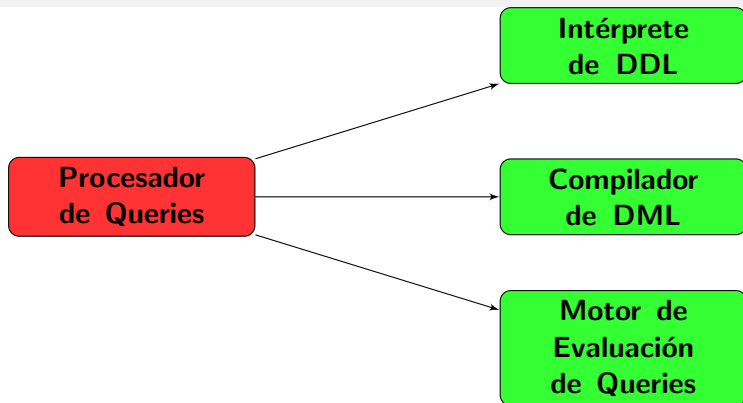
- ❶ Intérprete de DDL: diccionario de datos.
- ❷ Compilador de DML: plan de evaluación.
- ❸ Motor de evaluación: Instrucciones de bajo nivel del compilador de DML.

Procesador de Queries



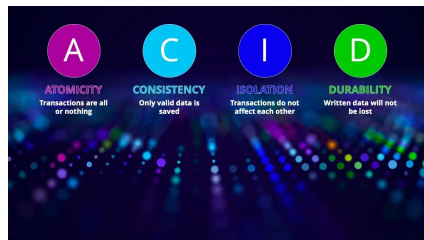
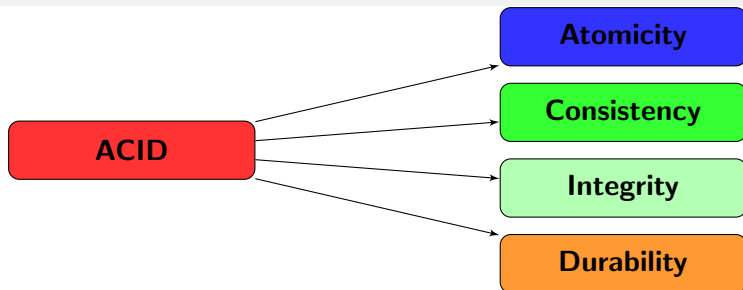
- ❶ Intérprete de DDL: diccionario de datos.
- ❷ Compilador de DML: plan de evaluación.
- ❸ Motor de evaluación: Instrucciones de bajo nivel del compilador de DML.

Procesador de Queries

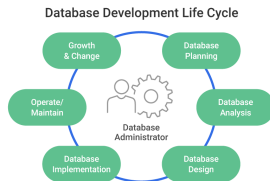
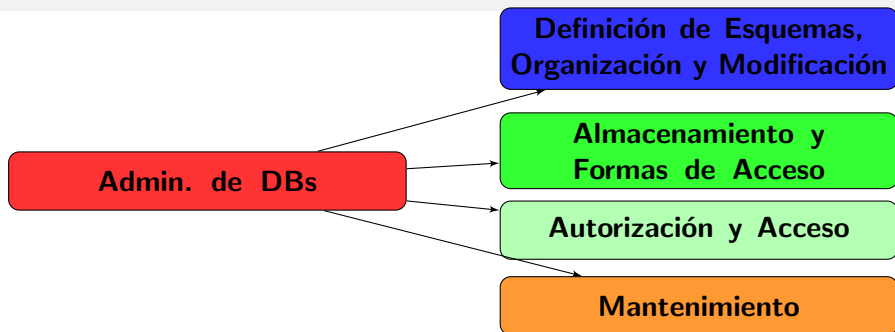


- ❶ Intérprete de DDL: diccionario de datos.
- ❷ Compilador de DML: plan de evaluación.
- ❸ Motor de evaluación: Instrucciones de bajo nivel del compilador de DML.

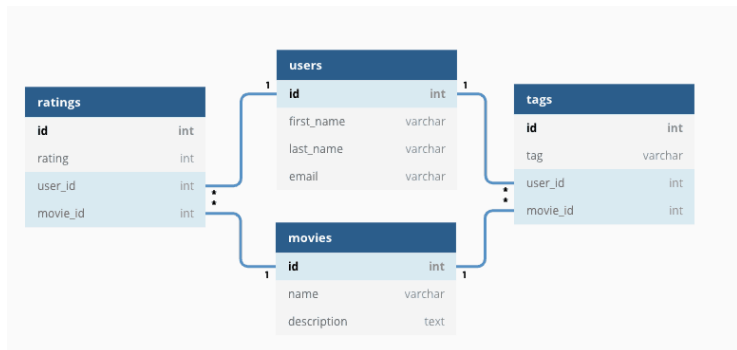
Propiedad ACID



Admin. de DBs

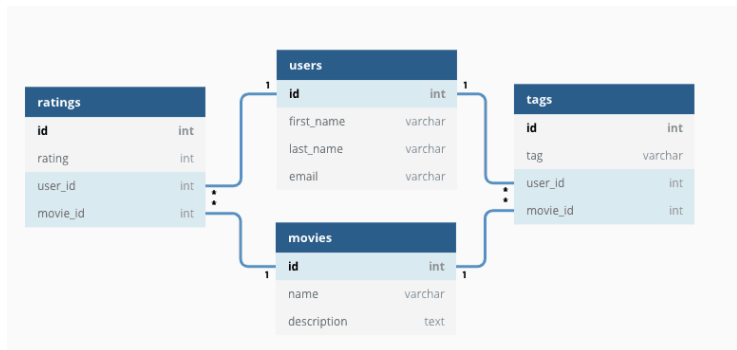


Estructura de una RDB



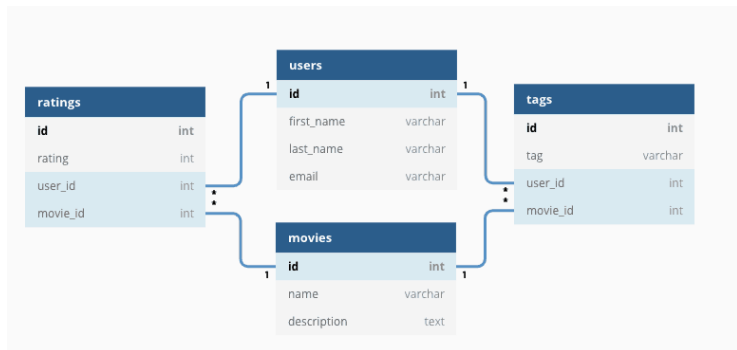
- ❶ Colección de tablas con nombre único.
- ❷ Tabla \rightarrow *Atributos*.
 - P. ej. tabla `users`: `id`, `first_name`, `last_name`
- ❸ Filas de n atributos/*tuplas*.

Estructura de una RDB



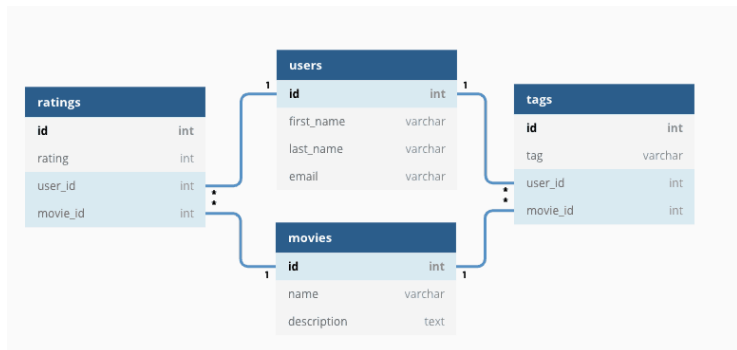
- ❶ Colección de tablas con nombre único.
- ❷ Tabla → *Atributos*.
 - P. ej. tabla users: id, first_name, last_name
- ❸ Filas de n atributos/*tuplas*.

Estructura de una RDB



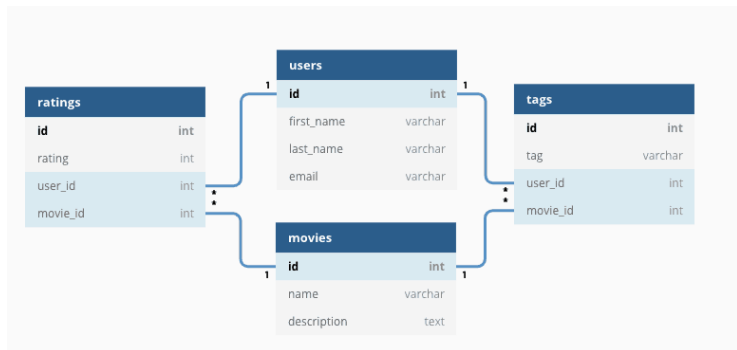
- ❶ Colección de tablas con nombre único.
- ❷ Tabla → *Atributos*.
 - P. ej. tabla **users**: id, first_name, last_name
- ❸ Filas de *n* atributos/*tuplas*.

Estructura de una RDB



- ❶ Colección de tablas con nombre único.
- ❷ Tabla → *Atributos*.
 - P. ej. tabla **users**: id, first_name, last_name
- ❸ Filas de *n* atributos/*tuplas*.

Estructura de una RDB



- ❶ Colección de tablas con nombre único.
- ❷ Tabla → *Atributos*.
 - P. ej. tabla **users**: id, first_name, last_name
- ❸ Filas de n atributos/*tuplas*.

Estructura de una RDB

Profesores:

ID	Apellido	Departamento	Salario (\$ mill)
10101	Srinivasan	Computación	15
12121	Wu	Finanzas	20
15151	Mozart	Música	8
22222	Feynman	Física	18
32343	El Said	Historia	9
76543	Crick	Biología	10

Cursos:

ID_curso	Nombre_curso	Departamento	Créditos
CS-315	Robótica	Computación	4
HIS-351	Historia Mundial	Historia	3
PHY-101	Principios de Física	Física	4
MU-199	Producción de videos musicales	Música	3

Estructura de una RDB

Profesores:

ID	Apellido	Departamento	Salario (\$ mill)
10101	Srinivasan	Computación	15
12121	Wu	Finanzas	20
15151	Mozart	Música	8
22222	Feynman	Física	18
32343	El Said	Historia	9
76543	Crick	Biología	10

Cursos:

ID_curso	Nombre_curso	Departamento	Créditos
CS-315	Robótica	Computación	4
HIS-351	Historia Mundial	Historia	3
PHY-101	Principios de Física	Física	4
MU-199	Producción de videos musicales	Música	3

Estructura de una RDB

I Requisitos:

ID_curso	ID_requisitos
CS-190	CS-101
CS-315	CS-101
BIO-301	BIO-101

II Dimensión tuplas en las tablas:

- Profesores: 4. Cursos: 4. Requisitos: 2.

III Instancias de las tablas:

- Profesores: 6 filas/datos. Cursos: 4 filas/datos. Requisitos: 3 filas/datos.

IV Resumiendo:

Relación: tabla.

Tupla: fila.

Atributo: columna.

Estructura de una RDB

I Requisitos:

ID_curso	ID_requisitos
CS-190	CS-101
CS-315	CS-101
BIO-301	BIO-101

II Dimensión tuplas en las tablas:

- Profesores: 4. Cursos: 4. Requisitos: 2.

III Instancias de las tablas:

- Profesores: 6 filas/datos. Cursos: 4 filas/datos. Requisitos: 3 filas/datos.

IV Resumiendo:

Relación: tabla.

Tupla: fila.

Atributo: columna.

Estructura de una RDB

I Requisitos:

ID_curso	ID_requisitos
CS-190	CS-101
CS-315	CS-101
BIO-301	BIO-101

II Dimensión tuplas en las tablas:

- Profesores: 4. Cursos: 4. Requisitos: 2.

III Instancias de las tablas:

- Profesores: 6 filas/datos. Cursos: 4 filas/datos. Requisitos: 3 filas/datos.

IV Resumiendo:

Relación: tabla.

Tupla: fila.

Atributo: columna.

Estructura de una RDB

I Requisitos:

ID_curso	ID_requisitos
CS-190	CS-101
CS-315	CS-101
BIO-301	BIO-101

II Dimensión tuplas en las tablas:

- Profesores: 4. Cursos: 4. Requisitos: 2.

III Instancias de las tablas:

- Profesores: 6 filas/datos. Cursos: 4 filas/datos. Requisitos: 3 filas/datos.

IV Resumiendo:

Relación: tabla.

Tupla: fila.

Atributo: columna.

Estructura de una RDB

I Requisitos:

ID_curso	ID_requisitos
CS-190	CS-101
CS-315	CS-101
BIO-301	BIO-101

II Dimensión tuplas en las tablas:

- Profesores: 4. Cursos: 4. Requisitos: 2.

III Instancias de las tablas:

- Profesores: 6 filas/datos. Cursos: 4 filas/datos. Requisitos: 3 filas/datos.

IV Resumiendo:

Relación: tabla.

Tupla: fila.

Atributo: columna.

Estructura de una RDB

I Requisitos:

ID_curso	ID_requisitos
CS-190	CS-101
CS-315	CS-101
BIO-301	BIO-101

II Dimensión tuplas en las tablas:

- Profesores: 4. Cursos: 4. Requisitos: 2.

III Instancias de las tablas:

- Profesores: 6 filas/datos. Cursos: 4 filas/datos. Requisitos: 3 filas/datos.

IV Resumiendo:

Relación: tabla.

Tupla: fila.

Atributo: columna.

Estructura de una RDB

I **Dominio:** valores permitidos para un atributo.

II **Relaciones (toda la DB):** $r, r_i, i \in \mathbb{N}$.

• Dominio *Atómico*: elementos *indivisibles*.

III **Atributo(s) en una relación:** A, B, C, \dots

IV En el caso de

ID	Apellido	Departamento	Salario (\$ mill)
10101	Srinivasan	CS	15
12121	Wu	FIN	20
15151	Mozart	MUS	8
22222	Feynman	PHYS	18
32343	El Said	HIS	9
76543	Crick	BIO	10

CS: ciencias de la computación, FIN: finanzas.

• ID *indivisible* \iff *elemento único e indivisible*.

Estructura de una RDB

❶ **Dominio:** valores permitidos para un atributo.

❷ **Relaciones (toda la DB):** $r, r_i, i \in \mathbb{N}$.

- Dominio *Atómico*: elementos *indivisibles*.

❸ **Atributo(s) en una relación:** A, B, C, \dots

❹ En el caso de

ID	Apellido	Departamento	Salario (\$ mill)
10101	Srinivasan	CS	15
12121	Wu	FIN	20
15151	Mozart	MUS	8
22222	Feynman	PHYS	18
32343	El Said	HIS	9
76543	Crick	BIO	10

CS: ciencias de la computación, FIN: finanzas.

- ID *indivisible* \iff *elemento único e indivisible*.

Estructura de una RDB

❶ **Dominio:** valores permitidos para un atributo.

❷ **Relaciones (toda la DB):** $r, r_i, i \in \mathbb{N}$.

- Dominio *Atómico*: elementos *indivisibles*.

❸ **Atributo(s) en una relación:** A, B, C, \dots

❹ En el caso de

ID	Apellido	Departamento	Salario (\$ mill)
10101	Srinivasan	CS	15
12121	Wu	FIN	20
15151	Mozart	MUS	8
22222	Feynman	PHYS	18
32343	El Said	HIS	9
76543	Crick	BIO	10

CS: ciencias de la computación, FIN: finanzas.

- ID *indivisible* \iff *elemento único e indivisible*.

Estructura de una RDB

❶ **Dominio:** valores permitidos para un atributo.

❷ **Relaciones (toda la DB):** $r, r_i, i \in \mathbb{N}$.

- Dominio *Atómico*: elementos *indivisibles*.

❸ **Atributo(s) en una relación:** A, B, C, \dots

❹ En el caso de

ID	Apellido	Departamento	Salario (\$ mill)
10101	Srinivasan	CS	15
12121	Wu	FIN	20
15151	Mozart	MUS	8
22222	Feynman	PHYS	18
32343	El Said	HIS	9
76543	Crick	BIO	10

CS: ciencias de la computación, FIN: finanzas.

- ID *indivisible* \iff *elemento único e indivisible*.

Estructura de una RDB

❶ **Dominio:** valores permitidos para un atributo.

❷ **Relaciones (toda la DB):** $r, r_i, i \in \mathbb{N}$.

- Dominio *Atómico*: elementos *indivisibles*.

❸ **Atributo(s) en una relación:** A, B, C, \dots

❹ En el caso de

ID	Apellido	Departamento	Salario (\$ mill)
10101	Srinivasan	CS	15
12121	Wu	FIN	20
15151	Mozart	MUS	8
22222	Feynman	PHYS	18
32343	El Said	HIS	9
76543	Crick	BIO	10

CS: ciencias de la computación, FIN: finanzas.

- ID **indivisible** \iff *elemento único e indivisible*.

Estructura de una RDB

❶ **Dominio:** valores permitidos para un atributo.

❷ **Relaciones (toda la DB):** $r, r_i, i \in \mathbb{N}$.

- Dominio *Atómico*: elementos *indivisibles*.

❸ **Atributo(s) en una relación:** A, B, C, \dots

❹ En el caso de

ID	Apellido	Departamento	Salario (\$ mill)
10101	Srinivasan	CS	15
12121	Wu	FIN	20
15151	Mozart	MUS	8
22222	Feynman	PHYS	18
32343	El Said	HIS	9
76543	Crick	BIO	10

CS: ciencias de la computación, FIN: finanzas.

- ID **indivisible** \iff *elemento único e indivisible*.

Esquemas de Tablas

- I Esquema de la DB: diseño lógico.
- II Instancia de la DB: contenido en cierto instante.

ID_curso	ID_requisitos
CS-190	CS-101
CS-315	CS-101
BIO-301	BIO-101

Esquema: 2 atributos. Instancia: 3 tuplas (modificables).

- IV Escritura del esquema:

nombre tabla(atributos) (1)

- V Tabla requisitos:

requisitos(ID_curso, ID_requisitos) (2)

Atributos: $A = \text{ID_curso}, \text{ID_requisitos}$.

Esquemas de Tablas

- I Esquema de la DB: diseño lógico.
- II Instancia de la DB: contenido en cierto instante.

ID_curso	ID_requisitos
CS-190	CS-101
CS-315	CS-101
BIO-301	BIO-101

Esquema: 2 atributos. Instancia: 3 tuplas (modificables).

- IV Escritura del esquema:

nombre tabla(atributos) (1)

- V Tabla requisitos:

requisitos(ID_curso, ID_requisitos) (2)

Atributos: $A = \text{ID_curso}, \text{ID_requisitos}$.

Esquemas de Tablas

- ❶ Esquema de la DB: diseño lógico.
- ❷ Instancia de la DB: contenido en cierto instante.

ID_curso	ID_requisitos
CS-190	CS-101
CS-315	CS-101
BIO-301	BIO-101



Esquema: 2 atributos. Instancia: 3 tuplas (modificables).

- ❸ Escritura del esquema:

nombre tabla(atributos) (1)

- ❹ Tabla requisitos:

requisitos(ID_curso, ID_requisitos) (2)

Atributos: $A = \text{ID_curso}, \text{ID_requisitos}$.

Esquemas de Tablas

- I Esquema de la DB: diseño lógico.
- II Instancia de la DB: contenido en cierto instante.

ID_curso	ID_requisitos
CS-190	CS-101
CS-315	CS-101
BIO-301	BIO-101

III

Esquema: 2 atributos. Instancia: 3 tuplas (modificables).

- IV Escritura del esquema:

nombre tabla(atributos) (1)

- V Tabla requisitos:

requisitos(ID_curso, ID_requisitos) (2)

Atributos: A = ID_curso, ID_requisitos.

Esquemas de Tablas

- ❶ Esquema de la DB: diseño lógico.
- ❷ Instancia de la DB: contenido en cierto instante.

ID_curso	ID_requisitos
CS-190	CS-101
CS-315	CS-101
BIO-301	BIO-101

❸

Esquema: 2 atributos. Instancia: 3 tuplas (modificables).

- ❹ Escritura del esquema:

nombre tabla(atributos) (1)

- ❺ Tabla requisitos:

requisitos(ID_curso, ID_requisitos) (2)

Atributos: $A = \text{ID_curso}, \text{ID_requisitos}$.

Esquemas de Tablas

❶ Tabla cursos:

ID_curso	ID_sección	Edificio	Salón	Departamento	Franja_horaria
BIO-101	1	Painter	514	BIO	B
CS-101	1	Packard	101	CS	H
CS-101	2	Packard	101	CS	F
CS-315	1	Packard	101	CS	D
PHY-101	1	Watson	100	PHY	A

Esquema: 6 atributos. Instancia: 5 tuplas.

❷ Franja_horaria: 8 opciones (A-H).

❸ Para esta tabla,

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)
(3)

❹ Atributo con dominio atómico: Franja_horaria.

Esquemas de Tablas

I Tabla cursos:

ID_curso	ID_sección	Edificio	Salón	Departamento	Franja_horaria
BIO-101	1	Painter	514	BIO	B
CS-101	1	Packard	101	CS	H
CS-101	2	Packard	101	CS	F
CS-315	1	Packard	101	CS	D
PHY-101	1	Watson	100	PHY	A

Esquema: 6 atributos. Instancia: 5 tuplas.

II Franja_horaria: 8 opciones (A-H).

III Para esta tabla,

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)
(3)

IV Atributo con dominio atómico: Franja_horaria.

Esquemas de Tablas

❶ Tabla cursos:

ID_curso	ID_sección	Edificio	Salón	Departamento	Franja_horaria
BIO-101	1	Painter	514	BIO	B
CS-101	1	Packard	101	CS	H
CS-101	2	Packard	101	CS	F
CS-315	1	Packard	101	CS	D
PHY-101	1	Watson	100	PHY	A

Esquema: 6 atributos. Instancia: 5 tuplas.

❷ Franja_horaria: 8 opciones (A-H).

❸ Para esta tabla,

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)
(3)

❹ Atributo con dominio atómico: Franja_horaria.

Esquemas de Tablas

I Tabla cursos:

ID_curso	ID_sección	Edificio	Salón	Departamento	Franja_horaria
BIO-101	1	Painter	514	BIO	B
CS-101	1	Packard	101	CS	H
CS-101	2	Packard	101	CS	F
CS-315	1	Packard	101	CS	D
PHY-101	1	Watson	100	PHY	A

Esquema: 6 atributos. Instancia: 5 tuplas.

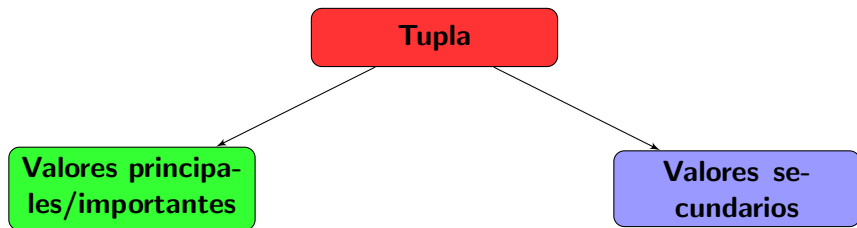
II Franja_horaria: 8 opciones (A-H).

III Para esta tabla,

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)
(3)

IV Atributo con dominio atómico: **Franja_horaria**.

Tablas y Claves



- II Esquema para una tabla *salones*:

salones(Edificio, Salón, Capacidad_máxima) (4)

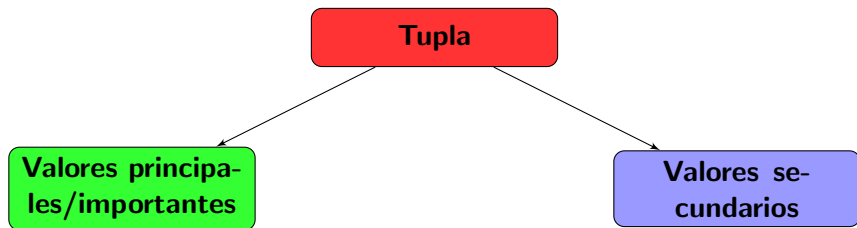
- III Valores **principales/importantes**: Edificio, Salón.

- IV Esquema para la tabla *profesores*:

profesores(ID, Apellido, Departamento, Salario (\$ mill)) (5)

- V Valor **principal/importante**: ID.

Tablas y Claves



Esquema para una tabla *salones*:

salones(Edificio, Salón, Capacidad_máxima) (4)



Valores **principales/importantes**: Edificio, Salón.



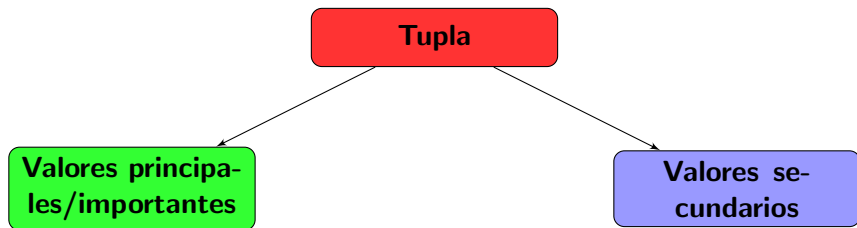
Esquema para la tabla *profesores*:

profesores(ID, Apellido, Departamento, Salario (\$ mill)) (5)



Valor **principal/importante**: ID.

Tablas y Claves



Esquema para una tabla *salones*:

salones(Edificio, Salón, Capacidad_máxima) (4)



Valores **principales/importantes**: Edificio, Salón.



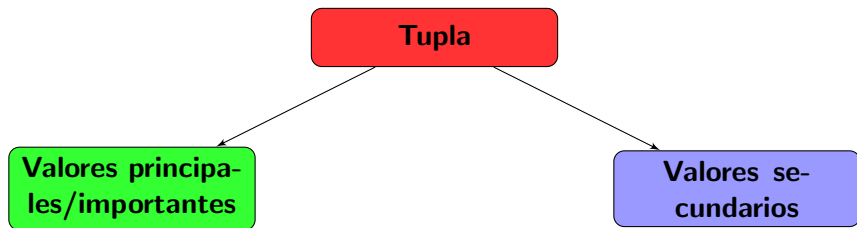
Esquema para la tabla *profesores*:

profesores(ID, Apellido, Departamento, Salario (\$ mill)) (5)



Valor **principal/importante**: ID.

Tablas y Claves



Esquema para una tabla *salones*:

salones(Edificio, Salón, Capacidad_máxima) (4)



Valores **principales/importantes**: Edificio, Salón.



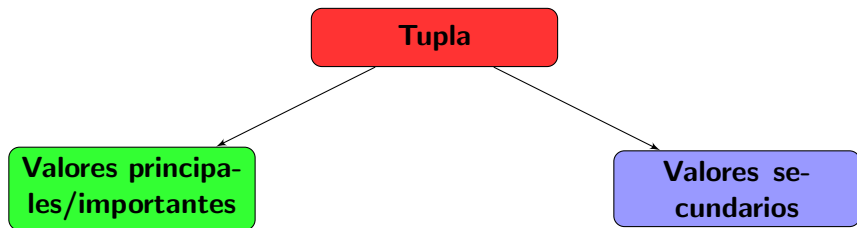
Esquema para la tabla *profesores*:

profesores(ID, Apellido, Departamento, Salario (\$ mill)) (5)



Valor **principal/importante**: ID.

Tablas y Claves



Esquema para una tabla *salones*:

salones(Edificio, Salón, Capacidad_máxima) (4)



Valores **principales/importantes**: Edificio, Salón.



Esquema para la tabla *profesores*:

profesores(ID, Apellido, Departamento, Salario (\$ mill)) (5)



Valor **principal/importante**: ID.

Tablas y Claves Foráneas

- ❶ Relación/tabla profesores:

profesores(ID, Apellido, Departamento, Salario (\$ mill)) (6)

- ❷ Relación/tabla cursos:

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria) (7)

- ❸ Relación/tabla requisitos:

requisitos(ID_curso, ID_requisitos) (8)

- ❹ Relación/tabla salones:

salones(Edificio, Salón, Capacidad_máxima) (9)

Tablas y Claves Foráneas

- ❶ Relación/tabla profesores:

profesores(ID, Apellido, Departamento, Salario (\$ mill)) (6)

- ❷ Relación/tabla cursos:

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria) (7)

- ❸ Relación/tabla requisitos:

requisitos(ID_curso, ID_requisitos) (8)

- ❹ Relación/tabla salones:

salones(Edificio, Salón, Capacidad_máxima) (9)

Tablas y Claves Foráneas

- ❶ Relación/tabla profesores:

profesores(ID, Apellido, Departamento, Salario (\$ mill)) (6)

- ❷ Relación/tabla cursos:

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria) (7)

- ❸ Relación/tabla requisitos:

requisitos(ID_curso, ID_requisitos) (8)

- ❹ Relación/tabla salones:

salones(Edificio, Salón, Capacidad_máxima) (9)

Tablas y Claves Foráneas

- ❶ Relación/tabla profesores:

profesores(ID, Apellido, Departamento, Salario (\$ mill)) (6)

- ❷ Relación/tabla cursos:

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria) (7)

- ❸ Relación/tabla requisitos:

requisitos(ID_curso, ID_requisitos) (8)

- ❹ Relación/tabla salones:

salones(Edificio, Salón, Capacidad_máxima) (9)

Tablas y Claves Foráneas

I Profesores y cursos:

profesores(ID, Apellido, Departamento, Salario (\$ mill))

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)

II Clave foránea: Departamento

III Cursos y requisitos:

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)

requisitos(ID_curso, ID_requisitos)

IV Clave foránea: ID_curso

Tablas y Claves Foráneas

I Profesores y cursos:

profesores(ID, Apellido, Departamento, Salario (\$ mill))

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)

II Clave foránea: **Departamento**

III Cursos y requisitos:

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)

requisitos(ID_curso, ID_requisitos)

IV Clave foránea: **ID_curso**

Tablas y Claves Foráneas

i Profesores y cursos:

profesores(ID, Apellido, Departamento, Salario (\$ mill))

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)

ii Clave foránea: **Departamento**

iii Cursos y requisitos:

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)

requisitos(ID_curso, ID_requisitos)

iv Clave foránea: ID_curso

Tablas y Claves Foráneas

I Profesores y cursos:

profesores(ID, Apellido, Departamento, Salario (\$ mill))

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)

II Clave foránea: **Departamento**

III Cursos y requisitos:

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)

requisitos(ID_curso, ID_requisitos)

IV Clave foránea: **ID_curso**

Tablas y Claves Foráneas

- ① Sean (r_1, A, t_1) y (r_2, B, t_2) con t_i : tupla en relación r_i .
- ② Clave foránea (de r_1 a r_2):

$$\forall t_1 \in r_1, \exists t_2 \in r_2. \quad (10)$$

Tablas y Claves Foráneas

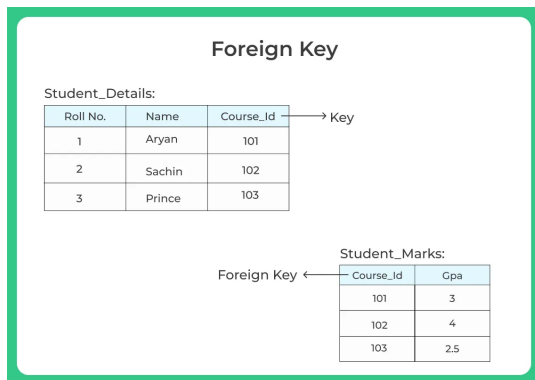
- ❶ Sean (r_1, A, t_1) y (r_2, B, t_2) con t_i : tupla en relación r_i .
- ❷ **Clave foránea** (de r_1 a r_2):

$$\forall t_1 \in r_1, \exists t_2 \in r_2. \quad (10)$$

Tablas y Claves Foráneas

- Sean (r_1, A, t_1) y (r_2, B, t_2) con t_i : tupla en relación r_i .
- Clave foránea** (de r_1 a r_2):

$$\forall t_1 \in r_1, \exists t_2 \in r_2. \quad (10)$$



Tablas y Claves Foráneas

❶ $\forall t_1 \in r_1, \exists t_2 \in r_2.:$ r_1 *referencia* a r_2 .

❷ Cursos y requisitos:

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)

requisitos(ID_curso, ID_requisitos)

- Cursos *referencia* a Requisitos.
- Requisitos *es referenciado* por Cursos.

❸ ID_curso: clave **principal** de requisitos.

- Ligadura de Clave Foránea/*Foreign-key Constraint*.

❹ **Ligadura de Integridad Referencial:**

Valores específicos de t_1 aparecen en valores específicos de t_2 . (11)

Tablas y Claves Foráneas

❶ $\forall t_1 \in r_1, \exists t_2 \in r_2.:$ r_1 *referencia* a r_2 .

❷ Cursos y requisitos:

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)

requisitos(ID_curso, ID_requisitos)

- Cursos *referencia* a Requisitos.
- Requisitos **es referenciado** por Cursos.

❸ ID_curso: clave **principal** de requisitos.

- Ligadura de Clave Foránea/*Foreign-key Constraint*.

❹ **Ligadura de Integridad Referencial:**

Valores específicos de t_1 aparecen en valores específicos de t_2 . (11)

Tablas y Claves Foráneas

❶ $\forall t_1 \in r_1, \exists t_2 \in r_2.:$ r_1 *referencia* a r_2 .

❷ Cursos y requisitos:

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)

requisitos(ID_curso, ID_requisitos)

- Cursos *referencia* a Requisitos.
- Requisitos **es referenciado** por Cursos.

❸ ID_curso: clave **principal** de requisitos.

- Ligadura de Clave Foránea/*Foreign-key Constraint*.

❹ **Ligadura de Integridad Referencial:**

Valores específicos de t_1 aparecen en valores específicos de t_2 . (11)

Tablas y Claves Foráneas

❶ $\forall t_1 \in r_1, \exists t_2 \in r_2.:$ r_1 *referencia* a r_2 .

❷ Cursos y requisitos:

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)

requisitos(ID_curso, ID_requisitos)

- Cursos *referencia* a Requisitos.
- Requisitos **es referenciado** por Cursos.

❸ ID_curso: clave **principal** de requisitos.

• Ligadura de Clave Foránea/*Foreign-key Constraint*.

❹ **Ligadura de Integridad Referencial:**

Valores específicos de t_1 aparecen en valores específicos de t_2 . (11)

Tablas y Claves Foráneas

❶ $\forall t_1 \in r_1, \exists t_2 \in r_2.:$ r_1 *referencia* a r_2 .

❷ Cursos y requisitos:

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)

requisitos(ID_curso, ID_requisitos)

- Cursos *referencia* a Requisitos.
- Requisitos **es referenciado** por Cursos.

❸ ID_curso: clave **principal** de requisitos.

- Ligadura de Clave Foránea/*Foreign-key Constraint*.

❹ Ligadura de Integridad Referencial:

Valores específicos de t_1 aparecen en valores específicos de t_2 . (11)

Tablas y Claves Foráneas

❶ $\forall t_1 \in r_1, \exists t_2 \in r_2.:$ r_1 referencia a r_2 .

❷ Cursos y requisitos:

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)

requisitos(ID_curso, ID_requisitos)

- Cursos referencia a Requisitos.
- Requisitos **es referenciado** por Cursos.

❸ ID_curso: clave **principal** de requisitos.

- **Ligadura de Clave Foránea/Foreign-key Constraint.**

❹ Ligadura de Integridad Referencial:

Valores específicos de t_1 aparecen en valores específicos de t_2 . (11)

Tablas y Claves Foráneas

❶ $\forall t_1 \in r_1, \exists t_2 \in r_2.:$ r_1 referencia a r_2 .

❷ Cursos y requisitos:

cursos(ID_curso, ID_sección, Edificio, Salón, Departamento, Franja_horaria)

requisitos(ID_curso, ID_requisitos)

- Cursos referencia a Requisitos.
- Requisitos **es referenciado** por Cursos.

❸ ID_curso: clave **principal** de requisitos.

- **Ligadura de Clave Foránea/Foreign-key Constraint.**

❹ **Ligadura de Integridad Referencial:**

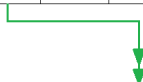
Valores específicos de t_1 aparecen en valores específicos de t_2 . (11)

Esquema de una DB

ID	Apellido	Departamento	Salario (\$ mill)
10101	Srinivasan	CS	15
12121	Wu	FIN	20
15151	Mozart	MUS	8
22222	Feynman	PHYS	18
32343	El Said	HIS	9
76543	Crick	BIO	10



ID_curso	ID_sección	Edificio	Salón	Departamento	Franja_horaria
BIO-101	1	Painter	514	BIO	B
CS-101	1	Packard	101	CS	H
CS-101	2	Packard	101	CS	F
CS-315	1	Packard	101	CS	D
PHY-101	1	Watson	100	PHY	A



ID_curso	ID_requisitos
CS-190	CS-101
CS-315	CS-101
BIO-301	BIO-101



→: clave foránea.



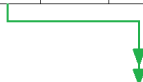
→→: Ligadura de Integridad Referencial.

Esquema de una DB

ID	Apellido	Departamento	Salario (\$ mill)
10101	Srinivasan	CS	15
12121	Wu	FIN	20
15151	Mozart	MUS	8
22222	Feynman	PHYS	18
32343	El Said	HIS	9
76543	Crick	BIO	10



ID_curso	ID_sección	Edificio	Salón	Departamento	Franja_horaria
BIO-101	1	Painter	514	BIO	B
CS-101	1	Packard	101	CS	H
CS-101	2	Packard	101	CS	F
CS-315	1	Packard	101	CS	D
PHY-101	1	Watson	100	PHY	A



ID_curso	ID_requisitos
CS-190	CS-101
CS-315	CS-101
BIO-301	BIO-101

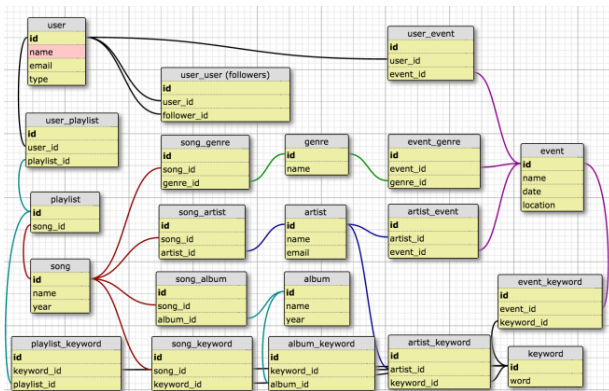


→: clave foránea.



→→: Ligadura de Integridad Referencial.

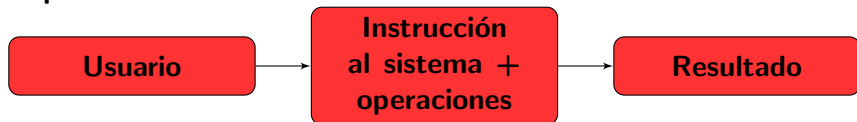
Esquema de una DB



- Tipos de relaciones: 1 a 1, 1 a muchas.

Lenguajes Relacionales

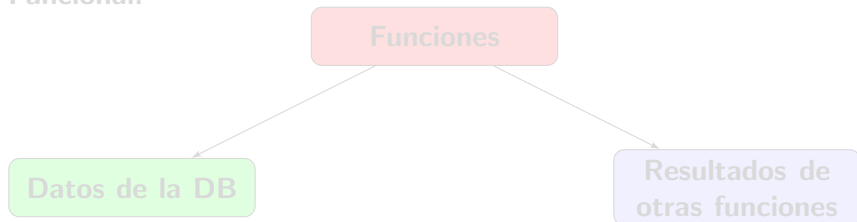
❶ Imperativo:



❷ Declarativo:

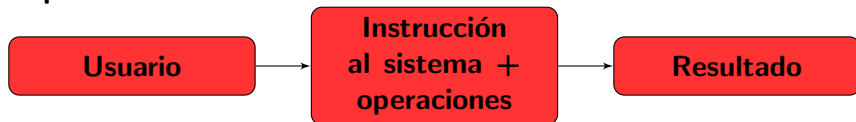


❸ Funcional:

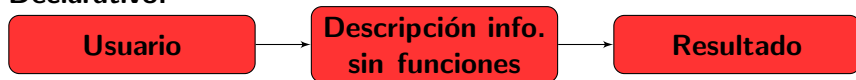


Lenguajes Relacionales

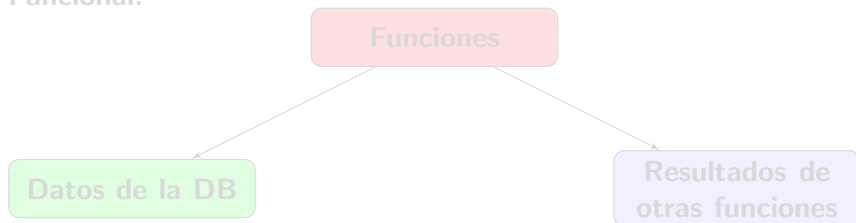
❶ Imperativo:



❷ Declarativo:

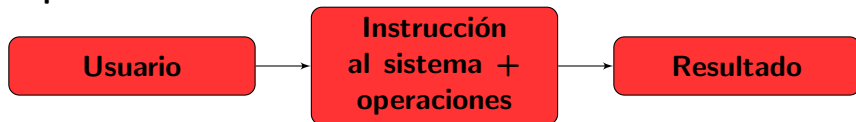


❸ Funcional:

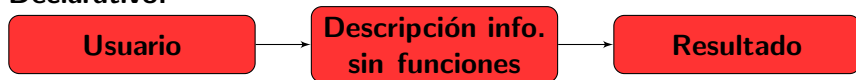


Lenguajes Relacionales

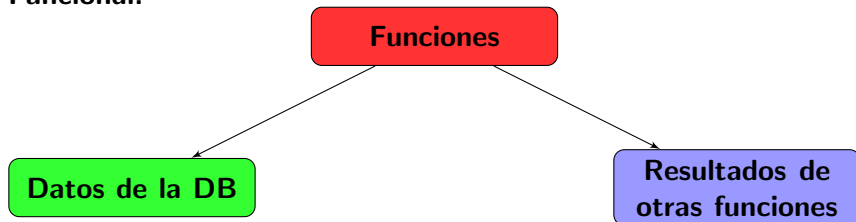
I Imperativo:



II Declarativo:



III Funcional:



Álgebra Relacional

- Operaciones f y g tal que

$$f(r_1) = r_2 \quad (\text{Unaria}),$$

$$g(r_1, r_2) = r_3 \quad (\text{Binaria}).$$

- Selección/SELECT: operación unaria.

$$\sigma_{\text{Atributo}} = \text{"Valor atributo"} (\text{relación}) \quad (12)$$

- Para la tabla profesores

ID	Apellido	Departamento	Salario (\$ mill)
10101	Srinivasan	CS	15
12121	Wu	FIN	20
15151	Mozart	MUS	8
22222	Feynman	PHYS	18
32343	El Said	HIS	9
76543	Crick	BIO	10

Álgebra Relacional

- Operaciones f y g tal que

$$f(r_1) = r_2 \quad (\text{Unaria}),$$

$$g(r_1, r_2) = r_3 \quad (\text{Binaria}).$$

- Selección/SELECT: operación unaria.

$$\sigma_{Atributo} = \text{"Valor atributo"} (\text{relación}) \quad (12)$$

- Para la tabla profesores

ID	Apellido	Departamento	Salario (\$ mill)
10101	Srinivasan	CS	15
12121	Wu	FIN	20
15151	Mozart	MUS	8
22222	Feynman	PHYS	18
32343	El Said	HIS	9
76543	Crick	BIO	10

Álgebra Relacional

- Operaciones f y g tal que

$$f(r_1) = r_2 \quad (\text{Unaria}),$$

$$g(r_1, r_2) = r_3 \quad (\text{Binaria}).$$

- Selección/SELECT: operación unaria.

$$\sigma_{Atributo} = \text{"Valor atributo"} (\text{relación}) \quad (12)$$

- Para la tabla profesores

ID	Apellido	Departamento	Salario (\$ mill)
10101	Srinivasan	CS	15
12121	Wu	FIN	20
15151	Mozart	MUS	8
22222	Feynman	PHYS	18
32343	El Said	HIS	9
76543	Crick	BIO	10

Álgebra Relacional (SELECT)

❱ $\sigma_{Salario = 18}$ (profesores) da como resultado

ID	Apellido	Departamento	Salario (\$ mill)
22222	Feynman	PHYS	18

❱ En cambio, $\sigma_{Salario < 18}$ (profesores)

ID	Apellido	Departamento	Salario (\$ mill)
10101	Srinivasan	CS	15
15151	Mozart	MUS	8
32343	El Said	HIS	9
76543	Crick	BIO	10

❱ $\sigma_{Apellido = Wu}$ (profesores)

ID	Apellido	Departamento	Salario (\$ mill)
12121	Wu	FIN	20

Álgebra Relacional (SELECT)

❱ $\sigma_{Salario = 18}$ (profesores) da como resultado

ID	Apellido	Departamento	Salario (\$ mill)
22222	Feynman	PHYS	18

❱ En cambio, $\sigma_{Salario < 18}$ (profesores)

ID	Apellido	Departamento	Salario (\$ mill)
10101	Srinivasan	CS	15
15151	Mozart	MUS	8
32343	El Said	HIS	9
76543	Crick	BIO	10

❱ $\sigma_{Apellido = Wu}$ (profesores)

ID	Apellido	Departamento	Salario (\$ mill)
12121	Wu	FIN	20

Álgebra Relacional (SELECT)

❱ $\sigma_{Salario = 18}$ (profesores) da como resultado

ID	Apellido	Departamento	Salario (\$ mill)
22222	Feynman	PHYS	18

❱ En cambio, $\sigma_{Salario < 18}$ (profesores)

ID	Apellido	Departamento	Salario (\$ mill)
10101	Srinivasan	CS	15
15151	Mozart	MUS	8
32343	El Said	HIS	9
76543	Crick	BIO	10

❱ $\sigma_{Apellido = Wu}$ (profesores)

ID	Apellido	Departamento	Salario (\$ mill)
12121	Wu	FIN	20

Álgebra Relacional (PROJECT)

- ❶ Proyección/Project (Operación Unaria):

$$\Pi_{\text{Atributos}}(\text{relación}). \quad (13)$$

- ❷ Resultado de $\Pi_{\text{ID, Apellido, Salario}}(\text{profesores})$:

ID	Apellido	Salario (\$ mill)
10101	Srinivasan	15
12121	Wu	20
15151	Mozart	8
22222	Feynman	18
32343	El Said	9
76543	Crick	10

Álgebra Relacional (PROJECT)

- ❶ Proyección/Project (Operación Unaria):

$$\Pi_{\text{Atributos}}(\text{relación}). \quad (13)$$

- ❷ Resultado de $\Pi_{\text{ID, Apellido, Salario}}(\text{profesores})$:

ID	Apellido	Salario (\$ mill)
10101	Srinivasan	15
12121	Wu	20
15151	Mozart	8
22222	Feynman	18
32343	El Said	9
76543	Crick	10

Composición de Operaciones

- Proyección de una selección:

$$\Pi_{\text{Atributos}}(\sigma_{\text{Atributo}}(r)). \quad (14)$$

- Ejemplos de composiciones:

$$\Pi_{\text{nombre}}(\sigma_{\text{Departamento} = \text{PHYS}}(\text{profesores}))$$

$$\Pi_{\text{nombre}, \text{salario}}(\sigma_{\text{Departamento} = \text{PHYS}}(\text{profesores}))$$

$$\Pi_{\text{ID_curso}, \text{ID_sección}}(\sigma_{\text{Departamento} = \text{BIO}}(\text{cursos}))$$

Composición de Operaciones

- Proyección de una selección:

$$\Pi_{\text{Atributos}}(\sigma_{\text{Atributo}}(r)). \quad (14)$$

- Ejemplos de composiciones:

$$\Pi_{\text{nombre}}(\sigma_{\text{Departamento} = \text{PHYS}}(\text{profesores}))$$

$$\Pi_{\text{nombre, salario}}(\sigma_{\text{Departamento} = \text{PHYS}}(\text{profesores}))$$

$$\Pi_{\text{ID_curso, ID_sección}}(\sigma_{\text{Departamento} = \text{BIO}}(\text{cursos}))$$

Operaciones Binarias: Producto Cartesiano

❶ Producto cartesiano:

$$r_{12} = r_1 \times r_2 \quad (15)$$

Operaciones Binarias: Producto Cartesiano

- ❶ Producto cartesiano:

$$r_{12} = r_1 \times r_2 \quad (15)$$

R

FName	LName	Age
John	Smith	35
Mary	Shelley	29
John	Doe	45
Nicky	Little	25

Cartesian Product

↓

R x S =

S

Car	Color
BMW	Black
Audi	White

FName	LName	Age	Car	Color
John	Smith	35	BMW	Black
John	Smith	35	Audi	White
Mary	Shelley	29	BMW	Black
Mary	Shelley	29	Audi	White
John	Doe	45	BMW	Black
John	Doe	45	Audi	White
Nicky	Little	25	BMW	Black
Nicky	Little	25	Audi	White

Operaciones Binarias: Producto Cartesiano

- ❶ Para dos relaciones con esquemas

profesor(ID, Apellido, Departamento, Salario),
 cursosdicta(ID, ID_curso, ID_sección)

- ❷ Los atributos de $\text{profesor} \times \text{cursosdicta}$ son

$$A = \text{profesor.ID, Apellido, Departamento, Salario, cursosdicta.ID, ID_curso, ID_sección} \quad (16)$$

donde $\text{profesor.ID} = \text{cursosdicta.ID}$.

- ❸ $N^\circ \text{ atributos total} = N^\circ \text{ atributos profesor} + N^\circ \text{ atributos cursosdicta}.$
- ❹ $N^\circ \text{ tuplas total} = N^\circ \text{ tuplas profesor} \times N^\circ \text{ tuplas cursosdicta}.$

Operaciones Binarias: Producto Cartesiano

- ❶ Para dos relaciones con esquemas

profesor(ID, Apellido, Departamento, Salario),
 cursosdicta(ID, ID_curso, ID_sección)

- ❷ Los atributos de profesor \times cursosdicta son

$$A = \text{profesor.ID, Apellido, Departamento, Salario, cursosdicta.ID, ID_curso, ID_sección} \quad (16)$$

donde profesor.ID = cursosdicta.ID.

- ❸ N° atributos total = N° atributos profesor + N° atributos cursosdicta.
- ❹ N° tuplas total = N° tuplas profesor \times N° tuplas cursosdicta.

Operaciones Binarias: Producto Cartesiano

- ❶ Para dos relaciones con esquemas

profesor(ID, Apellido, Departamento, Salario),
 cursosdicta(ID, ID_curso, ID_sección)

- ❷ Los atributos de profesor \times cursosdicta son

$$A = \text{profesor.ID, Apellido, Departamento, Salario, cursosdicta.ID, ID_curso, ID_sección} \quad (16)$$

donde profesor.ID = cursosdicta.ID.

- ❸ N° atributos total = N° atributos profesor + N° atributos cursosdicta.
- ❹ N° tuplas total = N° tuplas profesor \times N° tuplas cursosdicta.

Operaciones Binarias: Producto Cartesiano

- ❶ Para dos relaciones con esquemas

profesor(ID, Apellido, Departamento, Salario),
 cursosdicta(ID, ID_curso, ID_sección)

- ❷ Los atributos de $\text{profesor} \times \text{cursosdicta}$ son

$$A = \text{profesor.ID, Apellido, Departamento, Salario, cursosdicta.ID, ID_curso, ID_sección} \quad (16)$$

donde $\text{profesor.ID} = \text{cursosdicta.ID}$.

- ❸ $N^\circ \text{ atributos total} = N^\circ \text{ atributos profesor} + N^\circ \text{ atributos cursosdicta}$.
- ❹ $N^\circ \text{ tuplas total} = N^\circ \text{ tuplas profesor} \times N^\circ \text{ tuplas cursosdicta}$.

Operaciones Binarias: Join

❶ Sean (r_1, A, a_i) , (r_2, B, b_j) , donde

a_i : i -ésimo atributo de r_1 , con $a_i \in A$,

b_j : j -ésimo atributo de r_2 , con $b_j \in B$

❷ Join entre r_1 y r_2 :

$$r_1 \bowtie_{\theta} r_2 = \sigma_{\theta}(r_1 \times r_2) \quad (17)$$

- $\theta : r_1.a_i = r_2.b_j$.
- Ejemplo: en

profesor(ID, Apellido, Departamento, Salario),
 cursosdicta(ID, ID_curso, ID_sección),

$\theta = \text{profesor.ID} = \text{cursosdicta.ID}$.

Operaciones Binarias: Join

❶ Sean (r_1, A, a_i) , (r_2, B, b_j) , donde

a_i : i -ésimo atributo de r_1 , con $a_i \in A$,

b_j : j -ésimo atributo de r_2 , con $b_j \in B$

❷ Join entre r_1 y r_2 :

$$r_1 \bowtie_{\theta} r_2 = \sigma_{\theta}(r_1 \times r_2) \quad (17)$$

- $\theta : r_1.a_i = r_2.b_j$.
- Ejemplo: en

profesor(ID, Apellido, Departamento, Salario),
 cursosdicta(ID, ID_curso, ID_sección),

$\theta = \text{profesor.ID} = \text{cursosdicta.ID}$.

Operaciones Binarias: Join

❶ Sean (r_1, A, a_i) , (r_2, B, b_j) , donde

a_i : i -ésimo atributo de r_1 , con $a_i \in A$,

b_j : j -ésimo atributo de r_2 , con $b_j \in B$

❷ Join entre r_1 y r_2 :

$$r_1 \bowtie_{\theta} r_2 = \sigma_{\theta}(r_1 \times r_2) \quad (17)$$

- $\theta : r_1.a_i = r_2.b_j$.
- Ejemplo: en

profesor(ID, Apellido, Departamento, Salario),
 cursosdicta(ID, ID_curso, ID_sección),

$$\theta = \text{profesor.ID} = \text{cursosdicta.ID} .$$

Operaciones Binarias: Join

❶ Sean (r_1, A, a_i) , (r_2, B, b_j) , donde

a_i : i -ésimo atributo de r_1 , con $a_i \in A$,

b_j : j -ésimo atributo de r_2 , con $b_j \in B$

❷ Join entre r_1 y r_2 :

$$r_1 \bowtie_{\theta} r_2 = \sigma_{\theta}(r_1 \times r_2) \quad (17)$$

- $\theta : r_1.a_i = r_2.b_j$.
- Ejemplo: en

profesor(ID, Apellido, Departamento, Salario),
 cursosdicta(ID, ID_curso, ID_sección),

$\theta = \text{profesor.ID} = \text{cursosdicta.ID} .$

Otras Operaciones Binarias

Unión:

$$\Pi_{\text{Atributo(s)}}(\sigma_{\text{Condiciones 1}}(\text{relación})) \cup \Pi_{\text{Atributo(s)}}(\sigma_{\text{Condiciones 2}}(\text{relación})) \quad (18)$$

Condiciones: =, <, ≤, ...

Intersección:

$$\Pi_{\text{Atributo(s)}}(\sigma_{\text{Condiciones 1}}(\text{relación})) \cap \Pi_{\text{Atributo(s)}}(\sigma_{\text{Condiciones 2}}(\text{relación})) \quad (19)$$

- En SELECT: condición 1 \wedge condición 2, condición 1 \vee condición 2, etc.

Otras Operaciones Binarias

❶ Unión:

$$\Pi_{\text{Atributo(s)}}(\sigma_{\text{Condiciones 1}}(\text{relación})) \cup \Pi_{\text{Atributo(s)}}(\sigma_{\text{Condiciones 2}}(\text{relación})) \quad (18)$$

Condiciones: =, <, ≤, ...

❷ Intersección:

$$\Pi_{\text{Atributo(s)}}(\sigma_{\text{Condiciones 1}}(\text{relación})) \cap \Pi_{\text{Atributo(s)}}(\sigma_{\text{Condiciones 2}}(\text{relación})) \quad (19)$$

- En SELECT: condición 1 \wedge condición 2, condición 1 \vee condición 2, etc.

Otras Operaciones Binarias

❶ Unión:

$$\Pi_{\text{Atributo(s)}}(\sigma_{\text{Condiciones 1}}(\text{relación})) \cup \Pi_{\text{Atributo(s)}}(\sigma_{\text{Condiciones 2}}(\text{relación})) \quad (18)$$

Condiciones: =, <, ≤, ...

❷ Intersección:

$$\Pi_{\text{Atributo(s)}}(\sigma_{\text{Condiciones 1}}(\text{relación})) \cap \Pi_{\text{Atributo(s)}}(\sigma_{\text{Condiciones 2}}(\text{relación})) \quad (19)$$

- En SELECT: condición 1 \wedge condición 2, condición 1 \vee condición 2, etc.

Problemas

- ❗ Considere las siguientes DBs:

Empleados:

empleados(nombre_persona, dirección, ciudad)
trabajacon(nombre_persona, nombre_compañía, salario)
compañía(nombre_compañía, ciudad) (20)

Bancos:

oficina(nombre_oficina, ciudad_oficina, activos)
cliente(ID, nombre_cliente, dirección_cliente, ciudad_cliente)
préstamo(número_préstamo, nombre_oficina, cantidad)
prestatario(ID, número_préstamo)
cuenta(número_cuenta, nombre_oficina, balance) (21)

Problemas

- ❶ Encuentre las claves principales de cada relación en los esquemas.
- ❷ Encuentre las claves foráneas entre relaciones.
 - ¿Hay ligaduras de clave foránea?
 - ¿Hay ligaduras de integridad referencial?
- ❸ Haga el diagrama del esquema para cada una de estas DBs.

Problemas

1 Para la base de datos Empleados:

- Encuentre el nombre de cada empleado que vive en “Miami”.
- Encuentre el nombre de cada empleado con salario mayor a \$ 100000.
- Encuentre el nombre de cada oficina en la ciudad de “Miami”.
- Encuentre el nombre de cada empleado que cumpla con las dos primeras condiciones.

2 Para la base de datos Bancos:

- Encuentre el nombre de todas las oficinas que se encuentran en la ciudad de “Chicago”.
- Encuentre los números de préstamo con una cantidad mayor a \$10000.
- Encuentre el ID de cada depositario con un balance de cuenta mayor a \$ 6000.
- Encuentre el ID de todos los prestatarios que lo obtuvieron de la oficina “Downtown”.

3 Exprese sus resultados en operaciones de Álgebra Relacional.